

IOS – Instituto de
Oportunidade Social

HTML_CSS_JS 17 - Revisão Git e GitHub



- > Git
- > Comandos Git
- > GitHub
- > Criando repositório
- > Criando branches
- > Apagando branches

IOS – Instituto de
Oportunidade Social

Git



O que é Git?
Por que utilizamos o Git?



IOS – Instituto de
Oportunidade Social

Comandos Git



1° - No **gerenciador de arquivos** procure o local onde deseja iniciar seu repositório local.

- > Vamos criar uma pasta chamada ***Estudando Git;***
- > Entre na pasta;
- > Botão direito do mouse;
- > Selecione ***git bash here;***
- > Vai abrir o CMD do git;
- > Escreva ***git init*** (*Assim você iniciou um repositório local*);

Continuação na próxima página.

Por que utilizamos o git init ?

Para iniciar um repositório local para que seja possível a conexão com o repositório remoto (GitHub).

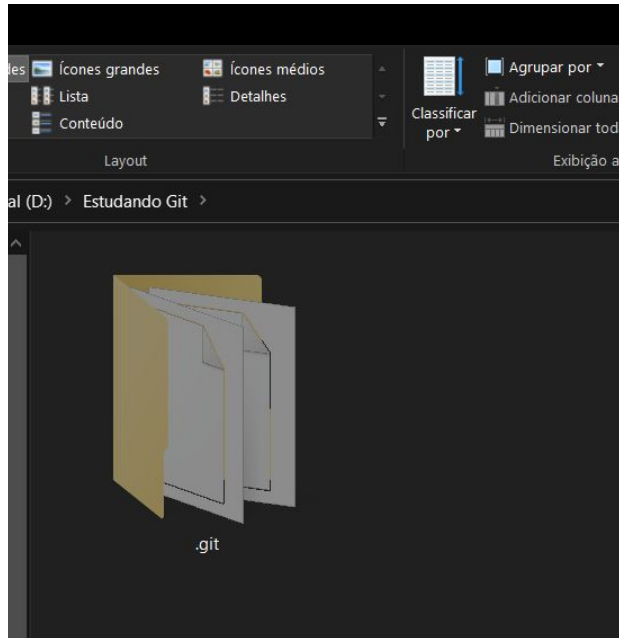
Antes do *git init*

```
MINGW64:/d/Estudando Git  
Heloisa Oliveira@MGNOTE0802 MINGW64 /d/Estudando Git  
$ git init|
```

Depois do *git init*

```
Heloisa Oliveira@MGNOTE0802 MINGW64 /d/Estudando Git (master)  
$
```

O que o *git init* gera na nossa pasta?



O *git init* cria uma pasta oculta chamada **.git**.

Para que serve o arquivo *.git* ?

- > git gerenciador do repositório
- > Histórico sobre todos os comandos feito no git, nome de branches, arquivos necessários para o gerenciador funcionar.


Configuração de E-mail e nome.

Você irá configurar o E-mail e nome de usuário que está cadastrado no seu GitHub

Configuração para o E-mail

```
MINGW64:/d/Estudando Git  
  
Heloisa Oliveira@MGNOTE0802 MINGW64 /d/Estudando Git (master)  
$ git config --global user.email "helloisa.beatriz538@gmail.com"
```

Configuração para o nome

 MINGW64:/d/Estudando Git

```
Heloisa Oliveira@MGNOTE0802 MINGW64 /d/Estudando Git (master)
$ git config --global use.name "Helloisa22"
```

Comando para verificar o nome e E-mail configurado

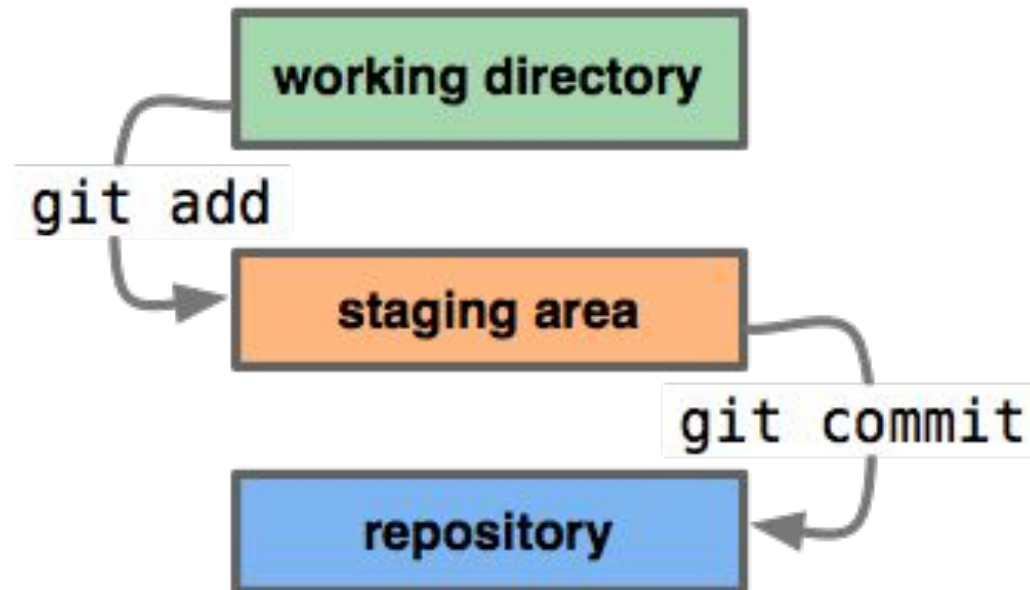
```
Heloisa Oliveira@MGNOTE0802 MINGW64 /d/Estudando Git (master)
$ git config user.name
Helloisa22
```

```
Heloisa Oliveira@MGNOTE0802 MINGW64 /d/Estudando Git (master)
$ git config user.email
helloisa.betriz44@gmail.com
```

- > Vamos criar um arquivo chamado **index.html**;
- > Com o git aberto digite o comando ***git status***;
- > O arquivo **index.html** tem que estar vermelho significando que o git pegou a nossa alteração que foi a inserção do arquivo **index.html**;
- > Significa que o arquivo não está na **staging area**;
- > No git digite o comando ***git add*** para adicionar as alterações no staging area;

O que é staging area ?

> Basicamente um espaço temporário onde você determina quais mudanças serão adicionadas no GitHub;



```
..\IOS> git status
```

```
On branch master
```

```
No commits yet
```

```
Changes to be committed:
```

```
(use "git rm --cached <file>..." to unstage)
```

```
new file:   app.js
```

```
new file:   index.html
```

→ staging area

```
Changes not staged for commit:
```

```
(use "git add <file>..." to update what will be committed)
```

```
(use "git restore <file>..." to discard changes in working directory)
```

```
modified:   index.html
```

```
..\IOS> █
```

→ Diretório local

Formas de adicionar arquivos no **staging area**

```
Heloisa oliveira@MGNO  
$ git add *|
```

```
Heloisa oliveira@MGNO  
$ git add *.html
```

```
Heloisa oliveira@MGNO  
$ git add .|
```

```
Heloisa oliveira@MGNO  
$ git add index.html
```

IOS – Instituto de
Oportunidade Social

Github





O que é Github?
Por que utilizamos o Github?

- > Crie um repositório público chamado ***Estudando Git;***
- > Copie o link HTTP para fazer a conexão do repositório remoto (Github) com o repositório local (Sua máquina);
- > No git digite o seguinte comando:
git remote add origin *linkHTTPdoGitHub*
- > Se não aparecer nenhuma mensagem de erro significa que foi configurado com sucesso.

Vamos executar o **commit**

```
Heloisa Oliveira@MGNOTE0802 MINGW64 /d/Est
Git (master)
$ git commit -m "Adicionei somente o JS"
```

Por que utilizamos o commit antes do push ?

> O commit é necessário para inserir um comentário de identificação sobre qualquer alteração feita no projeto.

Você pode ver o histórico de commits do repositório com o comando: ***git log***

```
MINGW64:/d/Estudando Git

Heloisa Oliveira@MGNOTE0802 MINGW64 /d/Estudando Git (master)
$ git log
commit 5ab33b3beabea90dd59ef342ec64166c3c637a56 (HEAD -> master, origin/master)
Author: Helloisa22 <helloisa.betriz44@gmail.com>
Date:   Wed Oct 20 00:14:54 2021 -0300

    Adicionei somente o JS

commit 6e74c39ec41f22fb5858b9d9efc9e9d85f284955
Author: Helloisa22 <helloisa.betriz44@gmail.com>
Date:   Tue Oct 19 23:47:07 2021 -0300

    testando caminhos
```

Por último vamos executar o comando **push** para enviar nossos arquivos ao Github:

> No git digite o comando:

git push -u origin *NomeDaSuaBranchAtual*

Atualize seu repositório e os novos arquivos estão adicionados no Github

IOS – Instituto de
Oportunidade Social

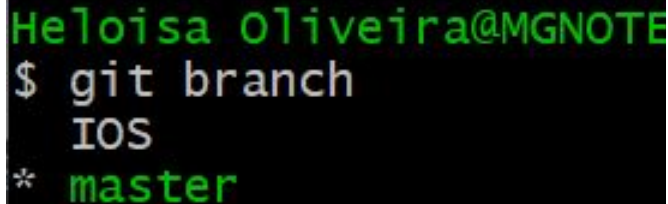
Criando Branches



> Comando para criar nova **branch** a partir da **branch** atual



```
MINGW64:/d/Estudando Git  
Heloisa Oliveira@MGNOTE08  
$ git branch IOS|
```



```
Heloisa Oliveira@MGNOTE  
$ git branch  
IOS  
* master
```

> Comando para listar todas as **branches** existente

- > Comando para navegar entre as **branches**

```
HeLoisa Oliveira@MGNOTE0  
$ git checkout IOS|
```

- > Comando para fazer o ***git merge*** entre duas **branch**

```
HeLoisa Oliveira@MGNOTE0802 MINGW64 /d/Estudando Git (master)  
$ git merge IOS -m "Adicionando a IOS a branch master"
```

> Comando para renomear a **branch**:

git branch -m <nome antigo> <nome novo>

```
HeLoisa Oliveira@MGNOTE0802 MING  
$ git branch -m master main|
```


- > Comando para clonar um repositório remoto
- > Selecione no repositório do Github o link HTTP
- > Em uma nova pasta, botão direito do mouse selecione Git Bashe Here
- > Digite **git clone** *linkHTTPdoGitHub*
- > O git fará o Download do seu repositório remoto
- > Ao finalizar digite o comando git pull para trazer definitivamente todos os arquivos do seu repositório remoto

IOS – Instituto de
Oportunidade Social

Apagando Branch



Para remover uma branch (ramo) local no Git, basta rodar o seguinte comando

```
HeLoisa Oliveira@MGNOTE0802 MINGW64 /d/Estudando Git (IOS)
$ git branch -d IOS
error: Cannot delete branch 'IOS' checked out at 'D:/Estudando Git'
```

```
HeLoisa Oliveira@MGNOTE0802 MINGW64 /d/Estudando Git (master)
$ git branch -d IOS
Deleted branch IOS (was 7c46920).
```

Comando para informar ao repositório remoto (Github) que você deletou uma **branch**

```
Heioisa Oliveira@MGNOTE0802 MINGW64 /d/Estudando Git (master)
$ git push origin :IOS
To https://github.com/Helloisa22/teste.git
- [deleted]          IOS
```

Atualize o seu repositório no Github e verifique a branch foi deletada

IOS – Instituto de
Oportunidade Social

Referências



Curso prático GIT e GITHUB - O que é git? - O que é GitHub? Porquê aprender git?

<https://www.youtube.com/watch?v=FF1f4bKYhoo&list=PLbEOwbQR9lqzK14I7OOeREEIE4k6rjgli>

w3schools

<https://www.w3schools.com/>